Differential I_{DDQ} Testable Static RAM Architecture

Waleed K. Al-Assadi
Anura P. Jayasumana
Yashwant K. Malaiya[†]

Technical Report CS-96-102

February 1996

Department of Electrical Engineering/
[†]Department of Computer Science
Colorado State University
Fort Collins, CO 80523

Differential I_{DDQ} Testable Static RAM Architecture *

W. K. Al-Assadi, A. P. Jayasumana, and Y. K. Malaiya†
Electrical Engineering Department
† Computer Science Department
Colorado State University
Fort Collins, CO 80523

Abstract

A testable design that enhances the I_{DDQ} testability of random access memories (SRAMs) for off-line testing is proposed. Increased accuracy and test speed can be achieved by memory array partitioning. Comaparision of I_{DDQ} values from two blocks is performed during parallel write/read operations to memory locations of the two blocks. Simultaneous write/read operations to all locations within physically interleaved block can significantly enhance the test speed as well as fault activation.

1 Introduction

With the increasing complexity of semiconductor memories, the nature of the failure modes have become more complex and subtle [1, 2]. Failure modes such as gate-oxide shorts, bridging defects, parasitic transistor leakage, defective p-n junctions, and transistors with incorrect threshold voltages, do not affect the logical behavior. Such faults may pass the functional and logical testing, but may malfunction overtime, causing reliability hazards. Many of those faults cause elevated quiescent power supply current (I_{DDQ}), which is typically several orders of magnitude greater than the I_{DDQ} of a fault-free device.

In static random-access memories (SRAMs), most of the I_{DDQ} testable faults are activated during the write/read cycles [3, 4, 5, 6]. In [3], an analysis of the effectiveness of the I_{DDQ} testing has been done using SRAM of 8K X 8-bit words manufactured by Philips, using Inductive Fault Analysis technique. The results show that a high fault coverage is achieved when I_{DDQ} testing is performed in combination with functional testing. In [7], experimental results were reported on deploying current testing to detect defects that cause data retention

*This work was supported partly by a BMDO funded project monitored by ONR.

problems. The idea of I_{DDQ} testing is expanded for fault localization in [6]. In [8], a testable SRAM structure was proposed for observing the internal switching behavior of the memory cells. The proposed structure provides a high coverage of disturb-type pattern sensitive faults. In [4], the detailed fault model of the 6-transistor memory cell was investigated for possible transistor level faults. It was shown that intra-cell defects can cause inter-cell faults in the memory array, such as coupling faults. Such faults were shown to cause elevated I_{DDQ} when activated. In [5] a testable design for memory array was shown to enhance I_{DDQ} testing by allowing parallel access to the whole memory cells during the write cycle.

The above work clearly establishes the promise of I_{DDQ} testing for SRAMs. However, it does not establish testability requirements that can improve the effectiveness of current testing. In addition the above work does not address the problem of current measurements and the effects of the size of the circuit under test on the accuracy and testing speed.

In this paper, we propose a testable scheme for off-line testing that enhances the I_{DDQ} testability for CMOS SRAMs. The proposed scheme partitions the memory array into identical partitions each with its own operational ground node (GND(O)). Quiescent power supply currents can be monitored during parallel access of locations or subset of locations (blocks) of each partition. Test speed is enhanced by comparing currents in the two partitions using a built-in current comparator (BICC). This paper is an extension of the differential concept proposed in [9]. In that work, a testable design for a word-organized SRAM architecture, which is a special case of SRAMs, was presented.

2 Differential I_{DDQ} Testing of SRAMs

The proposed scheme is intended to minimize the impact of the conventional built-in current sensor (BICS) circuit on speed testing, by using a current comparator. The memory array is partitioned physically into identical *partitions* each with its own operational ground node (GND(O)). The partitioning is done during the design phase. The number of partitions depend on the size of the memory array.

In most SRAM architectures, bits are physically separated such that individual bits belonging to several logical words may reside on a *segment* on the same chip, i.e. a segment i holds bit i of each word. Hence a partition in Figure 1 may represent a segment or several segments of the same chip, connected to a certain GND(O) node. GND(O) nodes are used during normal operations, thus bypassing the current comparator. During the test mode, the GND(O) nodes are open, and the test ground GND(T) node (ground node of the current comparators) will be the common ground node for the circuit , i.e. the tester will consider GND(T) as the ground node. Identical partition sizes imply similar ground line capacitance values for each partition.

A built-in current comparator (BICC) is used to compare the quiescent power supply

currents of two partitions during the testing mode while accessing their locations simultaneously. A sensor similar to the Differential BIC (D-BIC) proposed in [10] may be used as the BICC. With the D-BIC, testing is performed in two phases. The circuit is partitioned into two identical parts CUT1 and CUT2. Test vectors are applied to CUT1 and the current is compared to a reference current value in one phase. Then in the second phase, CUT2 is similarly tested. When using BICC, testing is performed in one phase, such that if the difference between the currents of two partitions (e.g. |I1-I2| in Figure 1) exceeds a suitably chosen I_{th}, the flag line is raised indicating a fault.

If several BICCs are used, the flag signals can be connected to an OR gate to generate a Pass/Fail signal. If the difference is less than I_{th}, no fault is detected. However, this may occur in case of faults driving similar currents in each partition. To overcome this problem, each partition can be divided into blocks. A *block* is a subset of contiguous or interleaved memory locations. Testing can be employed by simultaneously accessing two identical blocks, each belonging to a separate partition. Reducing the size of the accessed array will allow the resolution to be maintained. In addition, reducing the size of the partition will ensure that leakage currents will not add up to the point where they become comparable to the abnormal quiescent current expected of a fault. Therefore the accuracy and testing speed is expected to improve significantly.

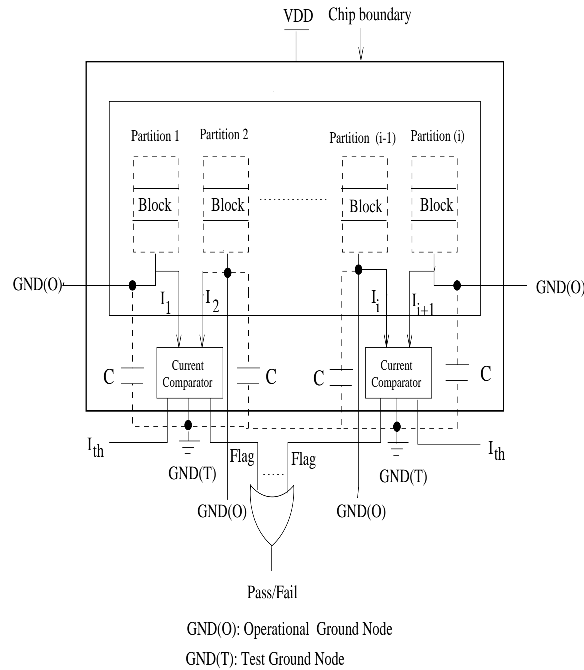

Figure 1: Principle of I_{DDQ} comparison

3 Partitioning and Decoder Design

In this section we propose a I_{DDQ} testable design for a SRAM architecture. Physical separation of the bits reduces the likelyhood of an alpha particle erasing more than one bit of a logical word, thus allowing effective single-bit error correction and enhances reliability. In this work we assume the the architecture shown in Figure 2. When the logical word length is 'd' bits, the memory array is composed of 'd' of segments. Each segment holds one bit of each logical word. A segment decoder is required to route input data to different segments.

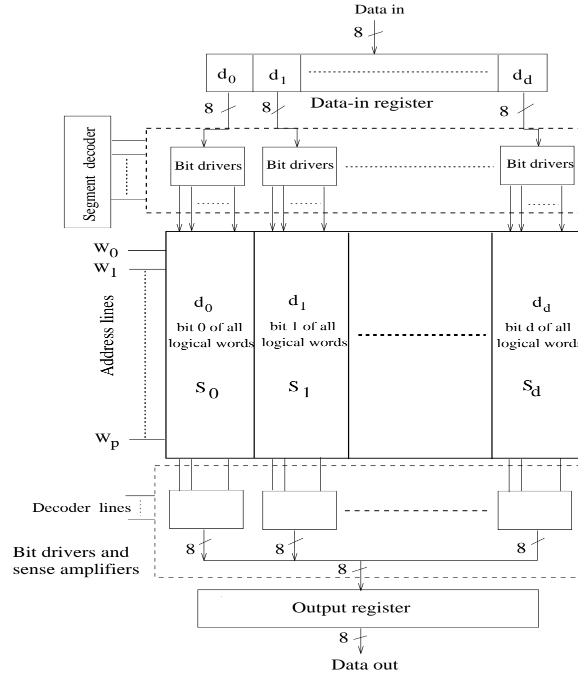

Figure 2: A SRAM Architecture

To achieve a I_{DDQ} testable scheme with high test accuracy and speed with small hardware overhead, design modifications are required to meet three basic goals [9]:

1 Efficient partitioning of memory array into identical blocks of a reasonable size.

2 Block write/read operations during the testing mode to access locations within a block in each partition in parallel.

3 Efficient BICC circuit with high sensitivity and accurate I_{th} selection.

The first and second goals are achieved by certain modifications to address and segment decoders, and memory array. Consider the memory array of 8K X 8-bit words shown in Figure 3. This array is composed of eight segments. The three most signifacent bits (a_{10}–a_{12}) of the row register are used for the segments decoder. The remaining bits ($a_0 - a_9$) are used for the 1K address decoder.

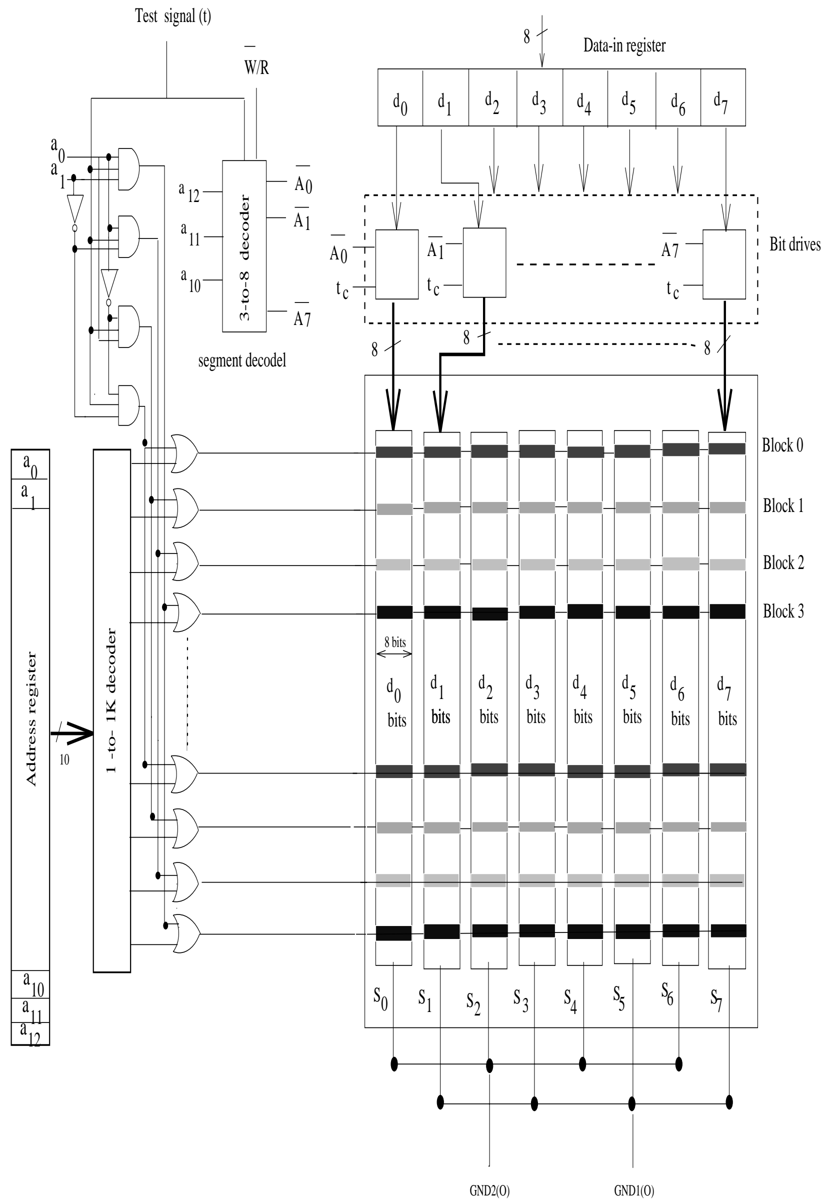

Figure 3: The I_{DDQ} 8K X 8 testable SRAM

If we are assuming only one BICC is used, i.e. the array can only be partitioned into two partitions, then two GND(O) nodes are required. Segments are divided equally among the GND(O) nodes. Segments S_0, S_2, S_4, and S_6 are connected to a one GND(O) node and the other segments are connected to the other GND(O) node. However, depending on the size of the memory array, more than two partitions may be required. If four partitions are to be used, then two segments are connected to a GND(O) node (i.e. four GND(O) nodes) and two BICCs are required. For the example under consideration, let us assume two partitions. A test signal (t) is required such that during normal operation mode t=0. During test mode when t=1, memory cells in both partitions are accessed simultaneously for

parallel write/read operations. The segment decoder is modified by adding the test signal (t). The write operation is performed when $\overline{W}/R$ signal is low.

The most crucial issue is the address decoder modifications that allow selective access to several physically contiguous or physically interleaved memory locations simultaneously. The conventional address decoder is modified to allow two modes of operation, normal and testing. To achieve high test speed, the process of current comparisons is needed to be limited to a minimum number, which implies minimum number of comparand blocks needed (goal 1 above). For the example under consideration, we assume that each of the 1K X 8 cell segment can only be divided into four blocks. In general it can be more than four blocks depending on the size of the memory array.

During the normal operation mode (t=0), the decoder works normally and is able to activate only one word line for each write or read operation. During the testing mode (t=1), the decoder is able to activate one block of locations simulatenously. To achieve this mode of operation, the two least significant bits a_0 and a_1 of the address register, and the test signal (t) are used to perform block selection during the testing mode as shown in Figure 3. Another test signal t_c is required to perform state-coupling test. Modifications to the bit drivers are shown in Figure 4. Bit drivers are modified such that when $t_c = 1$, test vectors of interchangeably complemented bits like (01010101) vector can be written to each physical word. Consider Figure 4 for the write operation to segment 0. When $t_c = 1$, decoder output lines $\overline{A_1}$, $\overline{A_3}$, $\overline{A_5}$, and $\overline{A_7}$ are activating their corresponding bit drivers to allow complement of the input bit to be written to the corresponding memory cells of the segment. This applied for each segment. However, the decoder works normally when $t_c = 0$.

In the test mode, each partition contains four blocks. Physical locations of the four blocks are interleaved with each other. The remaining higher order bits are to select the locations within the block in the normal mode of operation. The operation is such that during the testing mode, the AND gate that corresponds to a combination of (a_0, a_1) is active , thus the corresponding 1K address lines of the address decoder are active. These active lines of the address decoder are going to select 1K interleaved words (a block) in each partition. These two blocks are accessed simulatenously. The currents from both blocks are compared while parallel write/read operations are performed into both blocks. From this design shown in Figure 3, only one block is selected from each partition for current comaparision.

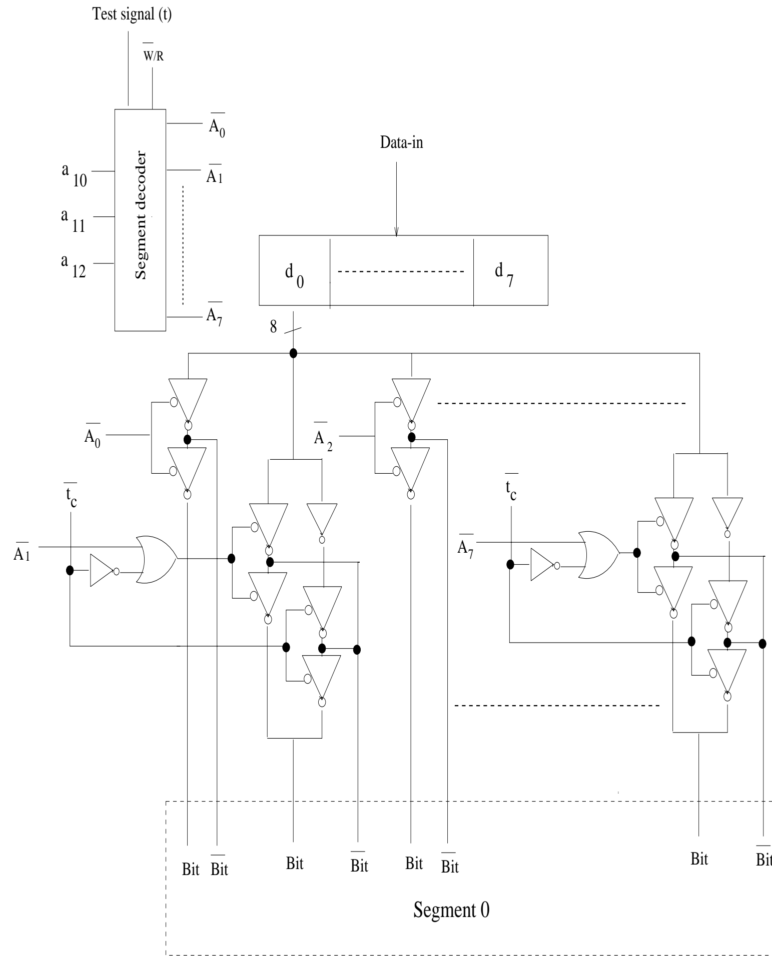

Figure 4: Bit-drive Modifications

4 Test Scheme

Faults that enhance I_{DDQ} are mainly transition faults, state coupling and bridging faults, and neighborhood pattern sensitive faults. To detect state coupling and bridging faults, all states of two adjacent cells i and j in a segment should be considered [11]. The testing sequence contains a set of parallel write/read operations to the blocks, such that if a test vector v is applied to *block* i, then $\overline{v}$ is applied to *block* $(i+1)$. A minimal test sequence contains two patterns each of (00000000) and (11111111) test vectors. The procedure is to apply a pattern without activating t_c signal. Then the same pattern is applied when t_c is activated. For example, when t_c is active and test vector (00000000) is applied, then vector (01010101) is written to the array. However, when vector (11111111) is applied while t_c is active, vector (10101010) will be written to the array.

This test sequence is capable of detecting stuck-at and a large fraction of state coupling and bridging faults. For each test vector applied, four block write operations and four block read operations are required as shown:

Write: block 0 $\leftarrow$ (00000000)
Write: block 1 $\leftarrow$ (11111111)
Write: block 2 $\leftarrow$ (00000000)
Write: block 3 $\leftarrow$ (11111111)
Read : block 0
Read : block 1
Read : block 2
Read : block 3

The above sequence is repeated for $t_c = 1$. If t_c is not used, it will take 8 times as long to write a pattern like (01010101) to a physical word. Although neighborhood pattern sensitive faults are considered complex faults and require a series of write/read operations into small sets of interleaved locations, the sequence above is capable of detecting some of those faults. From above, it is clear that 16 parallel write/read operations are required to test the SRAM for the faults assumed. This scheme may not cover some non I_{DDQ} testable failure modes which may need to be considered separately.

Several of the well known SRAM testing algorithms have a complexity proportional to n, where n is the number of memory locations. With the testable scheme proposed, the complexity of testing is proportional to b, where b is the number of blocks. Since $b \ll n$, the testing process is speeded-up proportional to (n/b). However the accuracy of testing depends also on the performance and sensitivity of the comparator BICS used for current monitoring.

5 Conclusions

We have extended the differential concept in [9] for a general I_{DDQ} testable SRAM design. The proposed scheme employs memory array partitioning and parallel write/read operations, during which several faults are activated with elevated quiescent power supply current. The currents are compared in one phase for each opearion. This enhances the testability such that testing can be performed in a significantly shorter time. A possible test sequence is presented. However, several questions remain unanswered in this area. For example, how to make optimal partitioning such that it will not add more hardware overhead, and accordingly how to make the selection of the of the threshold current I_{th}.

References

[1] C. A. Papachristou and N. B. Sahgal, “An Improved Method for Detecting Functional Faults in Semiconductor Random Access Memories,” *IEEE Trans. on Computers,* vol. c-34, no. 2, pp. 110-116, February 1985.

[2] R. Rajsuman and K. Rajkanan, "An Architecture to Test Random Access Memories," *Proc. 4th. Int'l. VLSI Design Conf.,* pp.144-147, Bangalore, India, January 1992.

[3] R. Meershoek, B. Verhest, R. McInerney and L. Thijssen, " Functional and I_{DDQ} Testing on a Static RAM," *Proc. Int'l Test Conf.,* pp.929-937, 1990.

[4] W. K. Al-Assadi, Y. K. Malaiya, and A. P. Jayasumana, "Modeling of Intra-Cell Defects in CMOS SRAM," *Records of the 1993 IEEE International Workshop on Memory Testing*, pp. 78-81, August 1993.

[5] H. Yokoyama, H. Tamamoto and Y. Narita, "A Current Testing for CMOS Static RAMs," *Records of the 1993 IEEE Int'l Workshop on Memory Testing,* pp. 137-142, August 1993.

[6] C. Elm and D. Tavangarian, "Fault Detection and Fault Localization Using I_{DDQ}-Testing in Parallel Testable FAST-SRAMs," *Proc. IEEE VLSI Test Symp.,* pp. 380-385, April 1994.

[7] C. Kuo, T. Toms, B. T. Neel, J. Jelemensky, E. A. Carter, and P. Smith, "Soft-Defect Detection (SDD) Technique for a High-Reliability CMOS RAM," *IEEE Journal of Solid-State Circuits,* vol. 25, no.1, pp. 61-66, February 1990.

[8] S. T. Su and R. Z. Makki, "Testing of Static Random Access Memories by Monitoring Dynamic Poer Supply Current," *Journal of Electronic Testing: Theory and Applications,* vol. 3, no. 3, August 1992, pp. 265-278.

[9] W. K. Al-Assadi, A. P. Jayasumana and . K. Malaiya, " A Bipartite, Differential I_{DDQ} Testable Static RAM Design," *Records of the 1995 IEEE International Workshop on Memory Technology, Design and Testing*, pp. 36-41, August 1995.

[10] A. D. Singh and J. P. Hurst, "Incorporating IDDQ Testing in BIST: Improved Coverage through Test Diversity," *Proc. IEEE VLSI Test Symp.,* pp.374-379, April 1994.

[11] R. Dekker, F. Beenker, and L. Thijssen, "A Realistic Fault Model and Test Algorithms for Static Random Access Memories," *IEEE Transaction on Computer-Aided Design,* vol. 9, no.6, June 1990, pp.567-572.